REMARKS

Claims 1, 15, 18 and 37 have been amended to clarify the subject matter regarded as the invention. Claim 45 is new. Claims 1, 3, 4, 7, 15, 18-20, 37, 39-41 and 45 are pending.

Claims 1, 15 and 37 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Hughes, U.S. Patent No. 6,122,372, in view of Yeager, U.S. Patent No. 6,167,402, in further view of Cloutier et al, U.S. Patent No. 6,535,586 (hereinafter "Cloutier").

The rejection is respectfully traversed. Neither Hughes, Yeager nor Cloutier teach retrieving messages from a first mailbox, computing message tags by concatenating the message sender and message sender submission time of the particular message and applying a hash algorithm to the concatenated string, then comparing message tags to messages from a second mailbox wherein the second mailbox is associated with a *different* recipient than the first mailbox (emphasis added) as recited in claims 1, 15, 37 and 45. Deduplicating mail across different mailboxes for different users includes the improvement of further efficiencies not taught in Hughes, Yeager and Cloutier.

Furthermore, the Examiner on pages 2-3 of the March 29, 2010 asserts:

> Applicant argues that Cloutier does not teach concatenating a message sender and a message sender submission time and applying a hash algorithm to the resulting string. Applicant further argues that Cloutier instead teaches hashing the from field and concatenating the result with the date field. However, Applicant's argument is incorrect.
>
> Cloutier discloses a "chksum" function that is used to generate a code signature from the date (message submission time) and from (sender) fields of a message [column 6 «lines 11-15»]. Within the chksum function, the date field information is placed into the "uid" variable as seen in the instruction at column 6, lines 25-28. The function proceeds to the next step and places the from field information into the "crunch" variable as seen in the instruction at column 6, lines 29-30. The two variables "uid" and "crunch" are then concatenated together with

the instruction "uid+=crunch«26". The instruction "+=" is the concatenating command.

The hashing function is then applied to the variable that contains the concatenated information from the date and from fields. – March 29, 2010 Office Action, emphasis added.

Applicant respectfully disagrees with Examiner's assertion that a hashing function is then applied to the variable that contains the concatenated information from the date and from fields. Cloutier on column 6, lines 11-15 discloses an algorithm that clearly:

1. First, Cloutier assigns to unsigned integer *uid* a value based on the "Date" field of an RFC 822 message; [column 6, lines 25-28]

2. Second, Cloutier assigns to unsigned integer *crunch* a value based on a classic XOR hashing function applied to the "From" field of an RFC 822 message; [column 6, lines 29-30]

   For example, if the "From" field is "Chris", *crunch* is assigned the value 67, by using the ASCII values for each character:

$$crunch \longleftarrow \mathrm{C} \oplus \mathrm{h} \oplus \mathrm{r} \oplus \mathrm{i} \oplus \mathrm{s}$$

$$crunch \leftarrow 67 \oplus 104 \oplus 114 \oplus 105 \oplus 115$$

$$crunch \leftarrow 67$$

   NB: *crunch* is an unsigned integer and does not contain the from field information as asserted in the March 29, 2010 Office Action, but contains the from field information as passed through the XOR hashing function.

3. Third, Cloutier assigns the concatenation of the unsigned integer *uid* representing the "Date" field with the unsigned integer *crunch* (which is not a string) representing the "From" field after the XOR hashing function to *uid*; [column 6, line 31]; and

4. Fourth, Cloutier allocates 25 bytes using the built-in malloc function to assign the concatenation stored in unsigned integer *uid_* to string *ret* using the built-in sprint function. [column 6, lines 32-33]

   NB: using the built-in malloc and sprintf functions to return a string value from an unsigned integer is not the same as applying a hash function as asserted in the March 29, 2010 Office Action.

Concatenating a message sender and a message sender submission time and applying a hash algorithm to the resulting string is **not** the same as applying a hash function to a From field and concatenating the result with a Date field. Support for the amendments in claims 1, 15, 37 and support for new claim 45 may be found in the above-captioned application without limitation in Figure 4 and paragraph 0038-0040. As such, claims 1, 15, 37 and 45 are believed allowable.

Claims 3, 4 and 7, which depend from claim 1; claims 18-20, which depend from claim 15; and claims 39-41, which depend from claim 37 are believed to be allowable for the same reasons described above.

The foregoing amendments are not to be taken as an admission of unpatentability of any of the claims prior to the amendments.

Reconsideration of the application and allowance of all claims are respectfully requested based on the preceding remarks. If at any time the Examiner believes that an interview would be helpful, please contact the undersigned.

Respectfully submitted,

Dated: August 29, 2010

/Christopher C. Tan/
Christopher C. Tan
Registration No. 61,950
V 408-207-4761
F 408-973-2595

VAN PELT, YI & JAMES LLP
10050 N. Foothill Blvd., Suite 200
Cupertino, CA 95014